

Smart camera with embedded co-processor : a postal sorting application

R. Mosqueron^a J. Dubois^b and M. Mattavelli^a

^aEcole Polytechnique Fédérale de Lausanne (EPFL), GR-LSM, CH 1015 Lausanne Switzerland

^bUniversité de Bourgogne, Laboratoire LE2I, 21000 Dijon, France

ABSTRACT

This work describes an image acquisition and processing system based on a new co-processor architecture designed for CMOS sensor imaging. The platform permits to configure a wide variety of acquisition modes (random region acquisition, variable image size, multi-exposition image) as well as high-performance image pre-processing (filtering, de-noising, binarisation, pattern recognition). Furthermore, the acquisition is driven by an FPGA, as well as a processing stage followed by a Nexperia processor. The data transfer, from the FPGAs board to the Nexperia processor, can be pipelined to the co-processor to increase achievable throughput performances. The co-processor architecture has been designed so as to obtain a unit that can be configured on the fly, in terms of type and number of chained processing (up to 8 successive pre-defined pre-processing), during the image acquisition process that is dynamically defined by the application. Examples of acquisition and processing performances are reported and compared to classical image acquisition systems based on standard modular PC platforms. The experimental results show a considerable increase of the performances. For instance the reading of bar codes with applications to postal sorting on a PC platform is limited to about 15 images (letters) per second. The new platform beside resulting more compact and easily installable in hostile environments can successfully analyze up to 50 images/s.

Keywords: Smart camera, image processing, co-processor, postal sorting

1. INTRODUCTION

Nowadays, *smart* cameras are more and more applied for their specific performances and their processing capabilities in different application fields. For several class of applications, different camera architectures that include embedded processing units have been developed. They can be further classified as follows: Cameras including an embedded ASIC, where fixed processing is executed. The level of flexibility of such architecture is quite limited since only a few processing parameters can be configured according to application constraints. Such systems can be considered as being similar to an artificial retina sensor. There also exist cameras coupled with an embedded processing unit (DSP, FPGA). The problem of such architectural solutions is that the processing capabilities remains usually very limited. Cameras with embedded coprocessors enable the implementation of more powerful processing due to the high degree of flexibility and to the clear task separation between the different units. The camera developed in this work belongs to this class of system architectures (Processor+FPGAs based co-processor). In parallel with this class of smart camera, two other class exist: artificial retinas cameras and standard camera interfaced with a computer, but limited in terms of processing or bandwidth.

For very high-performance image-processing applications an adaptive image acquisition stage is very often the key feature enabling the achievement of real-time performance and thus satisfy the demanding application requirements. The high pixel rate to be transferred to the central processing unit from the image sensor is often the main system bottleneck in terms of performance. Moreover, whenever such high pixel rate can be reduced according to the analysis of its own content (i.e. image portions can be discarded not being relevant for the application), the response time of the system is too slow to adapt the acquisition stage to the relevant image

Further author information: (Send correspondence to R. Mosqueron)

R. Mosqueron: E-mail: romuald.mosqueron@epfl.ch, Telephone: +41 (0)21 693 5688

J. Dubois: E-mail: julien.dubois@u-bourgogne.fr, Telephone: +33 (0)3 80 39 36 09

M. Mattavelli: E-mail: marco.mattavelli@epfl.ch, Telephone: +41 (0)21 693 69 84

sequence content because the transfer time from the sensor to the CPU unit is too large. Such problem can also be seen in terms of system costs, the (very large) bandwidth required is very often too expensive in terms of equipment and interfaces. Although we have assisted in the past to a continuous increase of processing performances of single core processors, this trend is approaching its end because of the difficulty to brake and approach the 4GHz barrier. In the meantime we observe now the wide availability of low-cost high-speed high-resolution sensors. This fact not only pushes the required processing performance to higher and higher levels so as to cover new demanding applications, but requires new architectural approaches to reduce the costs of the interfacing and processing stages that are now the real bottleneck of such systems.

The co-processing approach has been investigated in the last few years by several authors. Some works presented in literature are based on hardware co-processing designs specifically dedicated to a single application.¹⁻³ The performance improvements reported in literature are quite relevant, when comparing architectures with or without co-processor, those have the speed-up factor of several hundreds. Other authors have proposed generic systems whose property is the possibility to implement different algorithms on a co-processing based architecture.⁴ The performance of such implementations, in terms of speed-ups factors, can be higher than for some specific processing. In the class of "generic" co-processor units, only a few authors have mentioned the possibility to control the image acquisition stage simultaneously with the processing stage. Gorgon proposed a co-processor unit to control the acquisition stage of Charge Coupled Devices (CCD) sensor.⁵ Jung et al. presented a pre-processing unit to control CMOS sensor,⁶ but the achieved functionality operates only on the specific image corrections used to compensate physical limitation of the CMOS sensor. Although CMOS sensors present very attractive properties, no works presented in literature have shown that acquisition can be adapted to the processing providing a processing stage similar to the one we can find in "artificial retina" sensor approaches.⁷ This paper describes a co-processor unit design (COP) providing an interface for the full control of the sensor acquisition process driven from the main application CPU. The main processor and the co-processor are respectively in charge of the high-level tasks, the acquisition and processing decision imposed by the application, and the lower-level tasks, characterized by high level of processing regularity and parallelism. The co-processor implementation is based on a standard Field-Programmable Gate Array (FPGA) technology.

The first interesting result achieved by implementing this architecture is that relevant speed-up factors are obtainable for reconfigurable processing modules, thus providing enough flexibility in term of choice of processing and in terms of acquisition mode defined on the fly by the application itself (selection and preprocessing of any kind of area of interest). The second interesting result is that such on the fly adaptation of the acquisition mode yields a further bandwidth reduction for the transfer of the image data to the central CPU. This feature represents for some application a further speed-up in the overall system performance in terms of reduction of processing or increase of the achievable acquisition/processing frame rate.

To improve the efficiency of this system, we implement several processing to permit bar code reading in a postal sorting application. This application is an existing industrial application and we are able to compare the results. In fact, the parallelism of the system increase the number of result per second. For example, read letters are multiplied by 2 in this application. It proves its compactness and robustness too, in comparison with the actual PC system.

The paper is organized as follows: section 2 presents the co-processor platform and its architecture. Section 3 presents how the inclusion of processing into the acquisition loop enables to exploit the features and innovations of CMOS based imaging. Finally, results of a complete postal sorting application is presented in section 4.

2. SYSTEM ARCHITECTURE AND APPLICATIONS

Figure 1 illustrates the main architectural components of the camera with embedded co-processing stage. The system is composed of an embedded frame-grabber equipped, at different levels, of processing capabilities for the image acquired by the sensor. This system in its experimental configuration is made of a compact stack of 4 boards, enabling to easily interface various types of sensor/cameras and thus answering to various resolution and acquisition speed requirements in the most modular and economic way. The four boards include:

- the motherboard containing the main processor
- the communication board

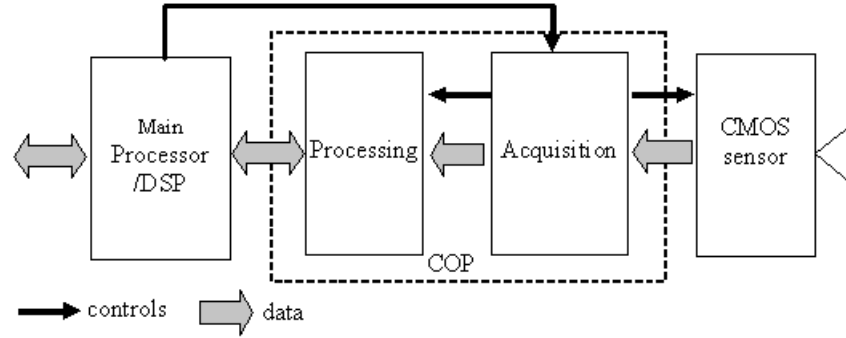


Figure 1. Block diagram of the co-processor based architecture.

- the board including the co-processor
- the camera interface board

The main additional advantages of this system, besides the capability of controlling the acquisition loop and the achievable processing performances compared to a traditional modular PC system, consist of:

- a low dissipated power,
- compact dimensions,
- a greater robustness (mean time between failures) because it does not integrate mobile components (ventilators, hard disks)
- a greater commercial lifespan because components in the computers world are very volatile and cannot be replaced with components having the same characteristics. Sometimes, after only a few years, partial redesign of the system is required to critical applications.

The mother board contains a Nexperia processor and includes functions for the sound and image processing. Around this DSP, we can find communication interfaces such as Ethernet, ISDN, etc., as well as acquisition and rendering of video images and analogical sound. The second board is based on a FPGA Spartan XL to manage the PCI arbiter, the communication interfaces such as USB2.0 and Firewire that can be driven to connect the camera with digital standard interfaces. On the third board, two FPGAs are used to acquire the pixels and to process the image coming from the camera sensor. The fourth board is a simple interface board between the FPGA board and the camera. A compact solution with only two boards is possible for low cost and compact solutions.

The main boards communicate through bus PCI v2.2 allowing to transfer a large number of data (upto 133Mbytes per second) to the host processor. However, the main idea of the system architecture is indeed to reduce as much as possible the data rate after the co-processor unit by transmitting only the processed image sections or by controlling the acquisition and to let the room on the bus for other interfaces such as the USB, the IEEE1394 that are supported and communicate with the host processor by the PCI bus.

This architectural solution provides exceptional processing potential and offers wide communication possibilities (RS-232, RS-485, USB, and Ethernet interface). The connectivity is achieved with a standard PCI bus. The co-processor unit is in charge of image acquisition and pre-processing. It implements a wide variety of acquisition modes (random region acquisition, variable image size, variable acquisition modes line/region based, multi-exposition image) and high-performance image pre-processing (calibration, filtering, de-noising, binarisation, pattern recognition). The pre-processing part is independent from the acquisition part. The processing part is built with pipelined or parallel HW processing modules to obtained high-performance. Furthermore, a processing and data transfer, from CMOS sensor to processor, can be operated in parallel to increase performance. Eventually, the co-processor architecture has been designed in order to as to obtain a unit that can be

configured on the fly, in terms of type and number of chained processing, during the image acquisition process that is defined by the application. A complete description is presented in a previous study.⁸

The system can be used with several CMOS image sensors, for the results described in this paper, a sensor IBIS4,⁹ with a resolution of 1280×1024 pixels and a 40 MHz pixel frequency has been used for the experimental results.

The essential problem of the co-processor architecture is the trade-off between processing efficiency and flexibility required to exploit the CMOS potential features. Two different parts essentially constitute the COP architecture (Fig. 2): the processing and the acquisition parts. The following functional blocks constitute the processing part: a processor interface (PCI interface), a command controller, a processing controller, a processing unit, a SRAM. The COP architecture is essentially constituted by the following functional blocks (Fig.2):

- a processor interface (bus interface),
- a bus bridge, a command controller,
- a processing controller,
- a processing structure,
- a CMOS sensor interface

The command controller receives the acquisition commands, the processing commands from the main application. The task scheduling is controlled by the processing controller and is executed by the processing structure unit configured according to the received commands. The data and image portions, provided by the main CPU and used by the co-processor for the actual processing tasks, are transferred to the processing structure via the bus bridge and via the processing controller. This feature enables to implement a true co-processing stage and not a simple pre-processing.

The link between sensor and acquisition part is specific for each image sensor, consequently it should be modified after any sensor change. The connection between acquisition and processing is standard, therefore independent of the sensor. Acquisition commands are constituted of parameters defined to cover a large number of acquisition modes to enable to interface a large sensor sort (linear CCD, CMOS matrix). Eventually, the connection with co-processor and processor are linked with standard PCI. Hence, the co-processor is independent of the processor and could be used as embedded IP with any PCI system. The co-processor architecture enables a full data rate to be obtained on PCI bus.

The possibility to adapt the number and nature of the processings and to operate on variable size/shape images is provided by the flexibility of the processing structure unit. A description of the global co-processor functionality is made in.¹² This system is adopting such principle, but an extension is made by adding new component to the system architecture.

The overall system can be described as an autonomous intelligent camera with powerful embedded processing when compared with modular systems associated with a computer.

The system has been thought for monitoring applications such as: road monitoring¹⁰ or intrusion detection or any other similar application. Quality control and control of industrial processes, where very high frame-rate on specific image sections are required, is another application field of the system. For such kind of processes, only the "relevant" portion of the images are necessary to be transmitted to the host CPU for further processing. In some cases only the result of the preprocessing, or of the processing (i.e. the detected feature), is needed to be transmitted outside the system to a local host PC or via Internet.

The association of the processing and acquisition stages aims at reducing the pixel rate for applications where "irrelevant" image portions are detected by the co-processor. The processing is then complemented by the Nexperia processor for higher level tasks at a possibly lower pixel-rate. The partition of the tasks is made by exploiting the specificity of each elements, to use it as efficiently as possible, thus reducing the pixel-rate when possible and the processing time so as to increase the overall throughput.

This architectural approach to the processes of sequence is particularly useful and performing, but not limited to the following application examples:

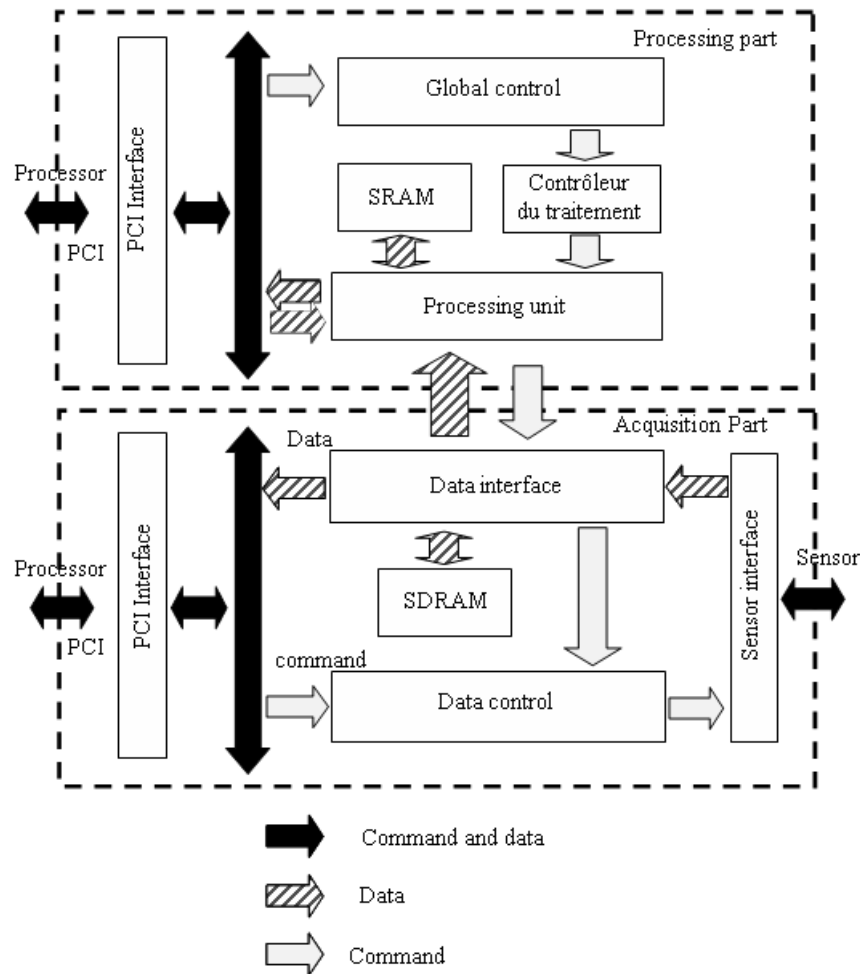


Figure 2. Block diagram of the COP architecture

- Tracking applications: to follow events of objects on a camera with transmission of the results to another camera which will resume the tracking,
- Pattern recognition applications: to recognize an object in a scene for which only a portion of the image needs to be further processed,
- Compression applications: to visualize or store sequences on a computer as for the video monitoring,
- Profilometry applications: where detection of objects profile depth and images need to be acquired on the same cameras.

Compression of video signals is generally used in camera systems to reduce the bandwidth of the data transfer and to be able to use a standard communication channel without addition of acquisition boards such as the camera-link for instance. However, with high performance sensors, there is immediately the problem of the connection that becomes now the system "bottleneck", and prevents from transferring the images rate provided by the sensor. The system described in this paper supports the implementation of a compression stage that makes it possible to approach to the sensor limit capabilities.¹¹

3. CO-PROCESSOR INTO PROCESSING ALGORITHM/ACQUISITION LOOP

The integration of a co-processing element into the image acquisition loop of a CMOS sensor has very interesting features. Standard CCD based image systems are synchronous and require that the full image is downloaded before proceeding to a new acquisition. CMOS sensors are much more flexible because not only they are intrinsically asynchronous, but they are also capable of performing image acquisitions on limited section of the sensor up to the acquisition of single pixels. For several applications such flexibility can be successfully exploited so as to reduce the data transfer to the central CPU thus considerably reducing the necessary data bandwidth. As consequence, the overall processing requirement of the application has just to process a limited portion of the original image. The key to achieve such results is to be able to provide to the main application the necessary information to adapt the acquisition stage without the need to transfer the full image to the central CPU. In other words, CMOS imaging can achieve:

- a selective image acquisition stage depending on the image content itself and on the requirements of the application,
- a relevant reduction of the data volume to be transmitted to the central CPU once the selective acquisition stage has been activated.

The condition for which such features can be achieved is that a "co-processing" element is inserted in the image acquisition loop driven by the "high level" application. In such architecture the "co-processing" unit besides the control of the acquisition stage becomes naturally in charge of the standard low-level repetitive tasks such as filtering, de-noising, binarisation, etc. In fact the full control of the acquisition stage enables the right control of the pre-processing tasks usually performed at the level of the central CPU or high-level application.

For instance, the "instructions" for a selective image acquisition stage, i.e. an acquisition stage for which only a (small) portion of the image that presents certain features need to be "acquired" and transmitted to the central CPU for further high level processing are handled by the "co-processor" accessing directly the CMOS sensor itself in an asynchronous manner. At this point also the processing associated to the specific feature "found" in the image can efficiently be implemented at the "co-processor" level. Then, only the "selected" image portion already pre-processed and/or pre filtered is transferred to the central CPU unit. The co-processing task schedule can be selected on the fly depending on the acquisition commands and is adapted to the acquisition form that is region/pixel based. With this architectural approach, only the CMOS sensor is providing the input image, thus the overall system results are very similar to an "artificial retina".⁷ By this approach the necessary data bandwidth can be drastically reduced eliminating in most of the cases the major system limitation. The main processor, freed from image acquisition and pre-processing tasks can then be used for further processing and/or high-level algorithms defined by the specific application.

The challenging aspects of the co-processor design are mainly related to the variable acquisition mode (i.e. input image format and layout). Obviously, the bandwidth associated to a window processing can be optimized, moreover the nature, the complexity and the number of possible processing stages can be adapted at each acquisition mode. In the examples of co-processing performance provided in this paper, the acquisition command word set generated by the processor are constituted essentially by two parts: the processing order with the parameters and the acquisition part. Each acquisition field is coded on 16 bits. Many different acquisition modes are then available. In all modes, a window can be selected in the full-range image, the size and the integration time are defined and moreover a sub-sampling (on Y and X) can also be specified. In simple multi-exposition mode, the same window is acquired several times or periodically and the delay between two acquisitions can be defined. Also in the tracking multi-exposition mode, the window can be translated. Such modes allow to create a "sub-image" image by row or column accumulation when the sensor is used as line sensor even with lines varying their position during the acquisition itself.

4. APPLICATION EXAMPLE : READING A BAR CODE FOR POSTAL SORTING

4.1 Application description

The postal sorting is a real-world example showing the processing possibilities and the achieved level of parallelism of the system. The goal of this application is to read bar codes on the letters, to enable automatic sorting at the

different stages of the logistic postal letter handling. If the bar codes cannot be read, the letter is rejected and need to be processed manually. This application has been developed with the objective of replacing an exiting platform which integrates a camera associated with a PC. The new embedded solution has been developed to increase as much as possible the processing performances and to obtain a portable and more flexible system.. Indeed due to the fact that bar codes printed on letters may be of bad quality or superposed to other visual information the possibility of implementing more complex processing increase the rate of correct detections/decodings achievable. Ideally, to correctly read the largest percentage of bar codes, each processing stage should require as much as possible processing power so as to guarantee that the bar code area is correctly localized (framed in the image 3). In reality the processing resources are limited and, results easier to extract and process a small part of the image that with a high probability includes the bar code, instead of dealing with the entire image of the letter which include extra information that can potentially create errors for the code bar detection and decoding. First, in the



Figure 3. Example of a bar code

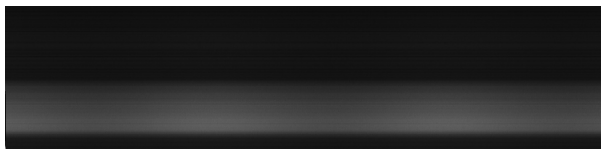
postal sorting application example, a letter is grabbed with the CMOS camera, the speed of the transporter is around 4 meters per second. Secondly, the system processes the image. For this application, in the co-processor platform, the used processings are:

- Transposition,
- Low pass filtering,
- Dilatation plus sub-sampling,
- Blobbing.

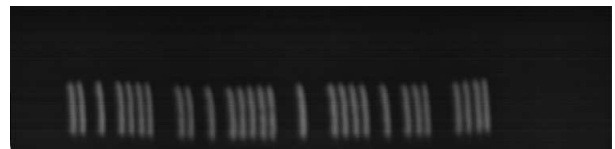
Blobbing task is performed in the processor. Details of these processing stages are provide in the following section. The final action is to read the bar code and send it to the postal sorting machine.

4.2 Details of the processing

To grab a letter, the CMOS camera is configured in a line scan mode. The camera is configured in this mode and not in an area scan mode due to the high speed of the transporter (4 meters per second), and the result image is deformed as shown in the image 4(a). However, the line scan mode is better suited to capture this kind of images. The camera grabs the same line during a predefined number of line or continuously and the acquisition FPGA rebuilds an image, this mode is shown in Figure 4(b). In this image, the difference between the two modes is shown, especially the effect on the bar code.



(a) Area scan mode



(b) Line scan mode

Figure 4. Acquisition mode difference

The first processing is a transposition. The transposition is used to rotate the rebuilt image to 90 degrees. A transposition is necessary because the other processing stages are specific to a horizontal reading (Figure 5). The first real image processing is a low pass filter. The low pass filter is used to delete the background and to raise

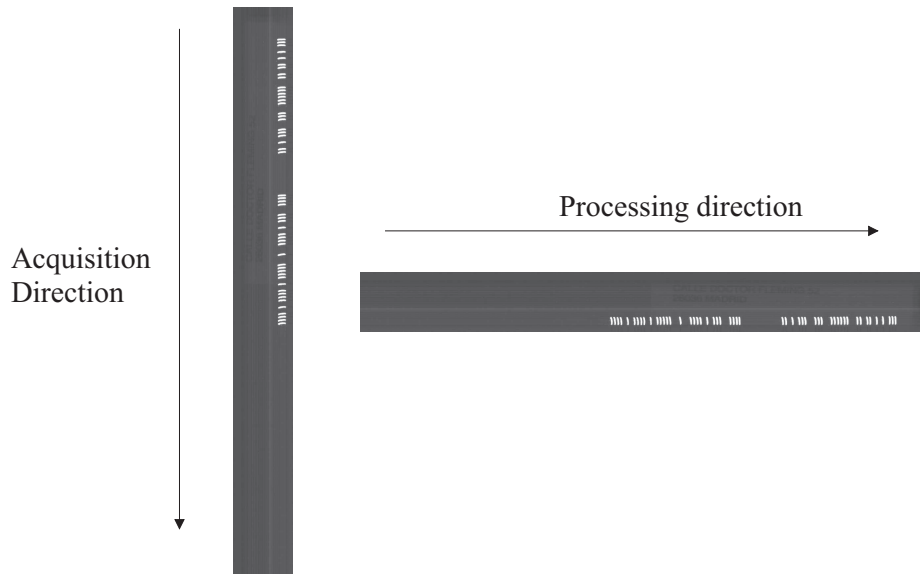
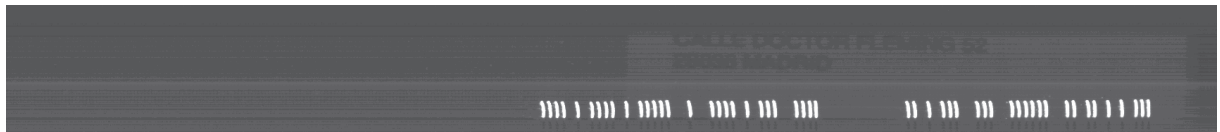
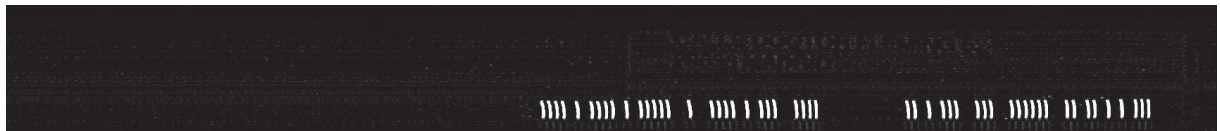


Figure 5. Transposition of an image

the white bar code as shown in Figure 6(b) compared to the original image in Figure 6(a). As shown previously, the data bus is a 32 bits bus, and transfers 4 pixels at the same time. The low pass filter is a convolution between the image and the window which includes the coefficients (11 coefficients is the best compromise for this action). The second processing is a dilatation. The dilatation is used to complete the region which integrates the bar



(a) Original image



(b) Low pass filtering



(c) Dilatation



(d) Sub-sampling



(e) Blobbing

Figure 6. Sequence into the co-processor

codes, thus it will be easier to detect this region as shown in Figure 6(c). This latter processing is performed to replace the central pixel by the maximum value of the 32 neighbor pixels. The sub-sampling is the third processing. In fact, only one of 4 pixels is transferred, moreover one line over four (Figure 6(d)). The goal is to divide the size of the image by 16 and consequently the original pixel bandwidth. These three last processing stages are performed in one dimension (line dimension) to obtain the best result and to reduce the processing time with the access of the second dimension. Therefore, the FPGA resource usage is reduced. As seen in the previous section, only the blobbing is made by the processor and all the other processing stages are performed by the FPGA (Co-processor). The blobbing is the last processing of the code bar detection. After the dilatation, several white (or grey) areas are designed. In the figure 6(d), two large areas are detected, that correspond to the area including the bar codes, but other areas can be detected which are probably not part of the code. The goal is to determinate the coordinates of the two zones that contain the code. So as to detect a region (blob), the image is described row by row and when a pixel superior to the threshold is founded, the object is squared and associated with a label. Once the image is fully analyzed and labelled, the two or three largest areas that are chosen probably include the bar code and the coordinates of these two objects, are extracted. Image 7 shows a part of the blobbing image where white areas are detected (squared in grey).



(a) Zoom of the sub-sampling image



(b) Zoom of the blobbing image

Figure 7. *Zoom of the blobbing processing*

The transfer of these coordinates is made to the co-processor and the co-processor transfers only the selected regions to the processor. The regions are taken on the filtering image which is stored temporally in the SRAM (Figure 8).



Figure 8. Bar code: zone transferred

When bar codes are transferred to the processor, the decoding can be activated. To decode the bar code, a FFT is made following several tests to read correctly percentage rates approaching 100% of bar code detected. As shown in Figure 8, the bar code is "1111010111101011111001001111010111001111" and after all the processing the system correctly read the code. In Figure 9, the efficacy of the system is illustrated also with a bad bar code which is not even readable at sight and without an appropriate processing. However, the system can correctly read it. Here, the bar code cannot be read exactly, but the system reads the correct bar code which is



(a) Original image not readable at sight



(b) Image after processing

Figure 9. *pre-processing for improvement of "unreadable" bar code*

”11100110110101011111001011101010111001111”. It proves the efficiency of the system.

So as to accelerate the decoding, the actions are performed in parallel and not in sequential order: acquisition (task 1), processing (task 2) and reading (task 3). These 3 tasks are executed in parallel to gain time and increase the number of letters processed. In Figure 10, it is a sequential sequence (Task 1 following 2 and 3) and during the acquisition the processor and the co-processor do not work. The same remarks are valid when the co-processor or the processor work.

The specificity of the platform is that the 3 principal components can work in parallel as shown in Figure 10

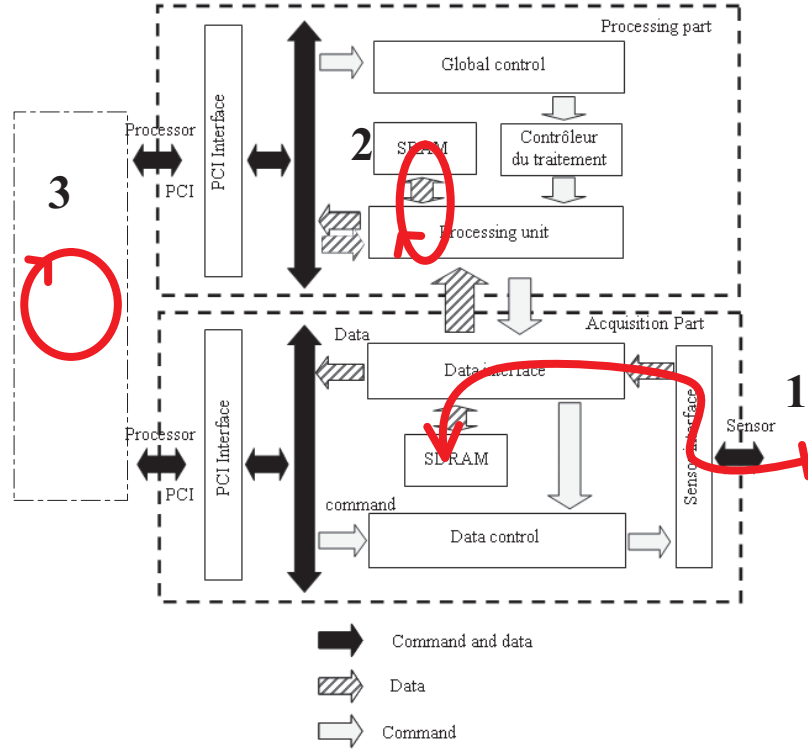


Figure 10. Sequential and parallel processing mode

(i.e. task 1 at the same time of 2 and 3). Only the transfer between the FPGA, prevents an acquisition or a processing into the co-processor. The processor is implemented in DMA mode and enables to work continuously. It receives data and at the same time it decodes the bar code. By using this configuration, the processing time is the same from acquisition to the output, but the number of processed letters is increased. Results are shown in the following section, and a comparison between a sequential mode, a parallel mode and the PC performances is made.

4.3 Results and comparisons

In this sub-section, results of the processings are provided. A comparison between the current system and the co-processor platform is made. The current system is a PC with the camera (BCi4 from Vector international¹³) associated with the compatible frame grabber. The co-processor platform is associated with the same camera. The current system is a PC with a processor 3.2 GHz and 1Go of RAM.

In the table 1, the necessary time for each processing needed to decode a bar code is shown. The tests were made with an image of 180 pixels width and 1712 rows captured. It is about a standard acquisition for a letter. Transfer is considered as a processing in the table. The transfer time from the co-processor to the processor by the PCI is considerably reduced. To transfer an entire image by the PCI, it takes 2.3 ms, but to transfer a sub-sampling image, it is 16 times less (0.15 ms) and for the bar code it is 20 times less (0.11 ms). This saving

Processing	Time (ms)
Acquisition	15.4
Transfer between the 2 FPGAs	4.6
Transposition	1.54
Low pass filtering	1.54
Dilatation plus sub-sampling	1.54
Transfer in the processor of the sub-sampling image	0.15
Blobbing	4
Transfer in the processor of the bar code image	0.11
Reading the bar code image	12
Total	40.88

Table 1. Processing time.

is very important factor to accelerate the processing.

The table 2 presents the comparison between :

- The co-processor platform and a sequential reading,
- The co-processor platform and a parallel reading,
- The actual PC.

	sequential	parallel	PC (ROI)
Time processing (ms)	40	40	40
Number of letter	15	30	15
Theoretical speed (m/s)	4	8	4

Table 2. Platform vs PC (approximative time)

In sequential and parallel mode, the processing time is around the same time as a PC, but with the parallelism, the number of the processed letters is increased (i.e. number of images). In the case of the PC, the size of the processed image is reduced to a small ROI (around 512×70), against 1712×180 with the co-processor platform. If the size is reduced to include correctly the bar code and not the image of the letter, the number of letters read can increase up to 50. Thus, the platform can read more images than the actual system and if it was possible to increase the speed of the physical transporter. Moreover, the co-processor platform is more efficient in hostile environment, small in size, and equivalent in term of the percentage of bar codes correctly read. The portability of the two systems is illustrated in the figure 11. The size is reduced and results more adequate for the integration industrial process.

5. CONCLUSION

Despite the increasing speed PC processors and bus frequency, the implementation of embedded co-processor systems expressly conceived for image sensors and inserted in the acquisition loop keeps several advantages. Very high processing speed and reduced image data bandwidth are achievable and maintain a high degree of flexibility in the pre-processing stage for the different acquisition modes specific of CMOS imaging. The described acquisition and co-processing embedded architecture is completely operational and the potential of such a new architecture are far from being fully exploited and are currently investigated in several challenging applications. The postal sorting application is a good example to show that the co-processor system increase the performances, reduce the consumption, compact and robust. The number of letters read is multiplied by 2 and more if we consider the same ROI (at least by 3). This system could be replace PC, in a lot a industrial applications, where the performances and the size are priority.



(a) Co-processor platform plus camera



(b) PC and co-processor platform

Figure 11. *Portability : Co-processor Platform Versus PC*

REFERENCES

1. B. Bosi, G. Bois, Y. Savaria : Reconfigurable pipelined 2-D convolvers for fast digital signal processing. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume 7, Issue 3, pp. 299–308, Sep 1999
2. C.W. Murphy, D.M. Harvey : Reconfigurable hardware implementation of BinDCT. Electronics Letters, Volume: 38 Issue 18, pp. 1012–1013, Aug 2002
3. N.W. Bergmann, Yuk Ying Chung : Video compression with custom computers. IEEE Transactions on Consumer Electronics, Volume 43, Issue 3, pp. 925–933, Aug 1997
4. C. Hinkelbein, et Al. : Pattern recognition algorithms on FPGAs and CPUs for the ATLAS LVL2 trigger. IEEE Transactions on Nuclear Science, Volume 47, Issue 2, pp 362–366, Apr 2000
5. M. Gorgon, J. Przybylo : FPGA based controller for heterogenous image processing system. Proceedings Euromicro Symposium on Digital Systems Design 2001, pp. 453–457, 2001
6. Yun Ho Jung, Jae Seok Kim, Bong Soo Hur, Moon Gi Kang : Design of real-time image enhancement preprocessor for CMOS image sensor. IEEE Transactions on Consumer Electronics, Volume 46 Issue 1, pp. 68–75, Feb 2000
7. F. Paillet, D. Mercier, T.M. Bernard : Second generation programmable artificial retina. Proceedings Twelfth Annual IEEE International ASIC/SOC Conference, pp. 304–309, 1999
8. R. Mosqueron, J. Dubois, M. Mattavelli : High Performance Embedded Co-Processor Architecture For CMOS Imaging Systems. Workshop on Design and Architectures for Signal and Image Processing (DASIP), Grenoble, 2007
9. Cypress : IBIS4 datasheet.
<http://www.chipcatalog.com/Datasheet/>
10. M. Bramberger, and Al. : Integrating Multi-Camera Tracking into a Dynamic Task Allocation System for Smart Cameras. IEEE Conference on Advanced Video and Signal Based Surveillance, pp. 474–479, Sept 2005
11. R. Mosqueron, J. Dubois, M. Paindavoine : High-speed camera with high resolution. EURASIP Journal of embedded systems, Volume 2007 Special Issue, In edition, 2007
12. J. Dubois, M. Mattavelli : Embedded co-processor architecture for CMOS based image acquisition. IEEE International conference of Image Processing, Volume 2, pp. 591–594, 2003
13. Vector international- CCAM Technologies.
<http://www.vector-international.be/>